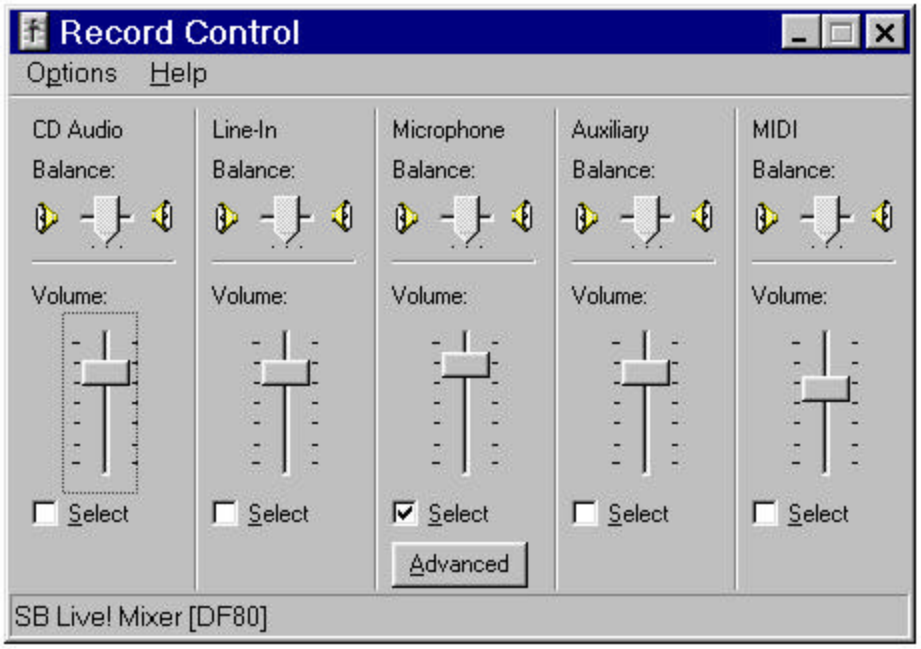
1.1 Matlab Recording & Analysis Program

Here we will present a Matlab program, *Record*, originally written by the author in the summer of 2001, and updated occasionally since. The program is intended to simplify the recording and basic editing of speech waveforms as well as to present the spectrogram and the time waveform in a side-by-side format for ease of analysis.

Getting Started:

Before you can do any recording through *Record*, you will need to connect a microphone or other sound source to the microphone input on your sound card. The next step is to ensure that your computer is set up to record from the microphone. On a Windows machine, you must select the microphone as the source in the *Record Control* window (see illustration below). The *Record Control* window can usually be accessed from a speaker icon in the system tray.



Microphone selected as the recording source

Windows' *Sound Recorder* program can be used to verify that the microphone is configured correctly. If sounds can be recorded using this program, they can also be recorded in Matlab. If you *can't* record sounds, there is some problem with the configuration.

Windows Sound Recorder in action

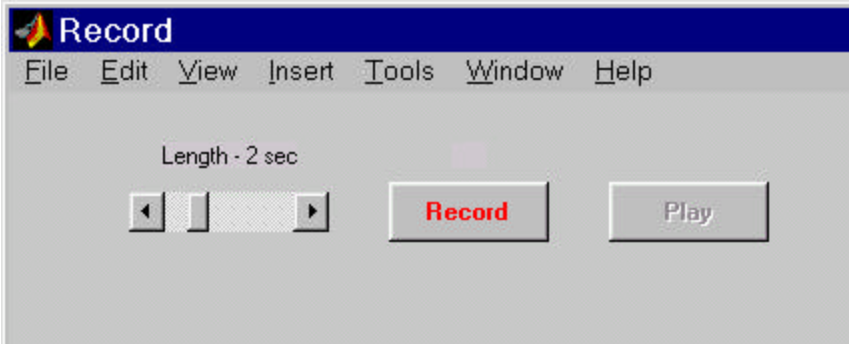
The remainder of this manual will describe the Matlab *Record* program—its inner working and functionality.

Running the program:

The program can be run by typing `record` at the Matlab prompt or by opening the program in the Matlab editor and selecting **Run** from the **Debug** menu

Recording:

Sound recording is initiated through the Matlab graphical user interface (GUI) by clicking on the record button. The duration of the recording can be adjusted to be anywhere from 1 to 6 seconds. (These are the GUI defaults, but the code can be modified to record for longer durations if desired.)



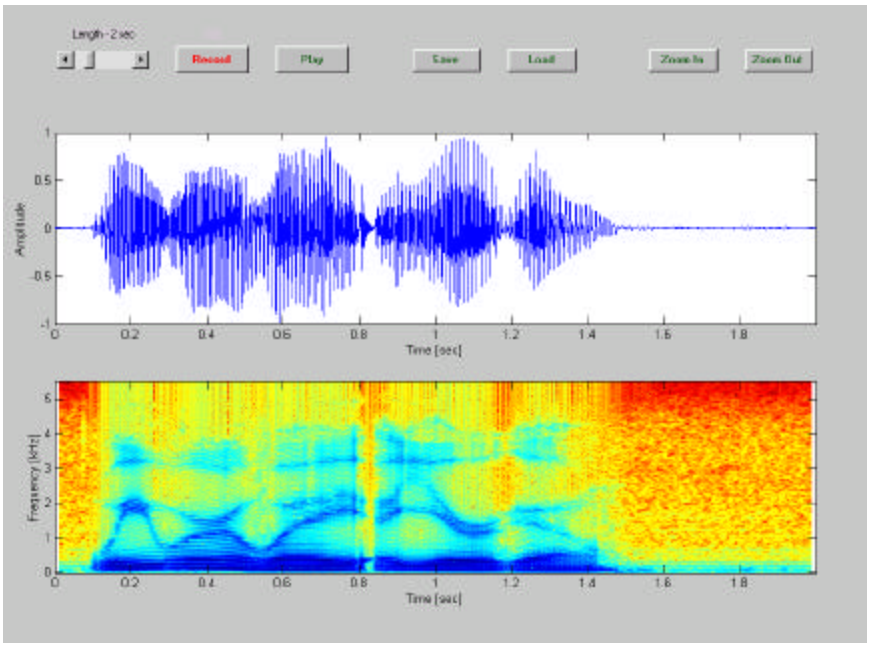
Upon being clicked, the record button executes a function that reads in mono data from the microphone jack on the sound card and stores it a Matlab vector.

Most of the important information in a typical voice waveform is found below a frequency of about 4 kHz. Accordingly, we should sample at a least twice this frequency, or 8 kHz. (Note that all sound cards have a built in pre-filter to limit the effects of aliasing.) Since there is at least some valuable information above 4 kHz, the Record GUI

has a default sampling rate of 16 kHz (however, the waveforms portrayed in this document were sampled at 11.025 kHz).

Once recorded, the time data is normalized to maximum amplitude of 0.99 and displayed on the upper plot in the GUI window. In addition to the time domain waveform, a spectrogram is computed using Matlab's built in `specgram` function (part of the signal processing toolbox).

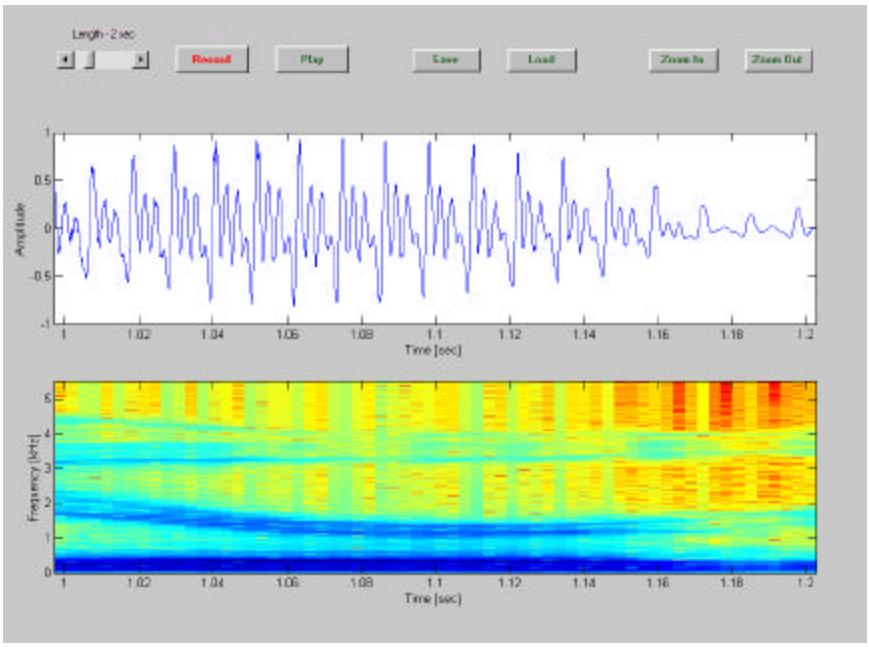
An example recording of the sentence, "We were away a year ago" is shown below.



"We were away a year ago"

Zooming in on the Waveform:

One can examine a region of interest in the waveform using the *Zoom in* button. When *Zoom in* is clicked, the cursor will change to a cross hair. Clicking the left mouse button and dragging a rectangle around the region of interest in the time domain waveform will select a sub-section of data. In the example below we have zoomed in on the region from about 1 to 1.2 seconds.



'Zoomed in' on the waveform

Zooming out:

The *Zoom out* button will change the axis back to what it was before *Zoom in* was used. If you zoom in multiple times, zooming out will return you to the previous axis limits.

Listening to the Waveform:

The *Play* button uses Matlab's `sound` function to play back (send to the speakers) the waveform that appears in the GUI. If you have zoomed in on a particular section of the waveform, only that portion of the waveform will be sent to the speakers.

Saving and Loading Data:

Save is used to write the waveform to a wave file. If zoomed in on segment of data, only that portion of the waveform will be saved.

Click *Load* to import any mono wave file into the Record GUI for analysis.

Record Program
Jesse Hansen
hansenj@ele.uri.edu